

## **Group Members:**

Ismael Barajas - ismaelbarajas30@csu.fullerton.edu

Patrick Mahoney - patrick.mahoney@csu.fullerton.edu

Zach Sarvas - zsarvas@csu.fullerton.edu

## **About:**

This project creates a Twitter-like microblogging service where users can post messages, repost messages, view global messages, see a timeline of messages from other users, narrow timelines to consist of only users they are following, follow or unfollow users, update user profile info, and see who you are following and who is following you. This service is implemented using two RESTful back-end services which make use of SQL databases and http basic authentication.

## **How To Run:**

- Replace your haproxy.cfg file with the provided file found in the “etc” folder called “haproxy.cfg”.
- Run ./bin/init.sh from the command line to create the databases.
- Run ./bin/foremanStart.sh from the command line to start foreman.
- Run sudo systemctl restart haproxy.

After following the above steps, the two microservices are now running as users.localhost and timelines.localhost, respectively.

# API Documentation:

- **User Microservice (URL: users.localhost):**

- **authenticated(auth\_user):**

- **Description:** This API is for validating the user login. If the username and password match the values in the database, we return the value “true”. If the validation returns false, we return 401 Status code which is “Unauthorized error”.
- **Uses basic authentication**
- **Endpoint:** /user/authenticate
- **Method:** GET
- **Parameters:** username, password
- **Sample Request:**

```
http -a ProfAvery:password users.localhost/user/authenticate
```

- **Success Response:**

```
HTTP/1.1 200 OK
content-length: 122
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:00:13 GMT
server: gunicorn/20.0.4
www-authenticate:

[
  {
    "bio": "wow does this work",
    "email": "kavery@fullerton.edu",
    "id": 1,
    "password": "password",
    "username": "ProfAvery"
  }
]
```

- **Error Response:**

```
HTTP/1.1 401 Unauthorized
content-length: 101
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:02:09 GMT
```

```
server: gunicorn/20.0.4
www-authenticate: basic realm="simple"

{
  "errors": {
    "Invalid Authentication": "Provided Basic HTTP Authentication
credentials were invalid"
  }
}
```

○ **user(username):**

- **Description:** Takes one parameter username, where username is the desired account information. GETs the account information for the given username.
- **Endpoint:** /user/<username>
- **Method:** GET
- **Parameters:** username (text)
- **Sample Request:**

```
http GET users localhost/user/ProfAvery
```

■ **Success Response:**

```
HTTP/1.1 200 OK
content-length: 122
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:04:36 GMT
server: gunicorn/20.0.4

[
  {
    "bio": "wow does this work",
    "email": "kavery@fullerton.edu",
    "id": 1,
    "password": "password",
    "username": "ProfAvery"
  }
]
```

■ **Error Response:**

```
HTTP/1.1 404 Not Found
content-length: 65
content-type: application/json; charset=utf-8
```

```
date: Sat, 23 Oct 2021 04:05:16 GMT
server: gunicorn/20.0.4

{
  "message": "ProfAver does not exist",
  "status": "404 Not Found"
}
```

○ **update\_user(user\_id, column, content):**

- **Description:** Takes three parameters user\_id, column, and content, where user\_id is the desired account, the column is the desired information to change, and content is the updated content. PUTs the updated account information into the table and removes the old content.
- **Endpoint:** /user/update
- **Method:** PUT
- **Parameters:** user\_id (number), column (text [username, email, password, bio]), content (text)
- **Sample Request:**

```
http PUT users.localhost/user/update/ user_id=2 column=username
content=newUsername
```

■ **Success Response:**

```
HTTP/1.1 200 OK
content-length: 60
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:10:16 GMT
server: gunicorn/20.0.4

{
  "message": "Your username has been updated to newUsername"
}
```

■ **Error Response:**

```
HTTP/1.1 400 Bad Request
content-length: 134
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:16:21 GMT
server: gunicorn/20.0.4
```

```
{
  "message": "The one or all of the inputs you provided is invalid, check documentation for correctness.",
  "status": "400 Bad Request"
}
```

○ **delete\_user(user\_id):**

- **Description:** Takes one parameter user\_id, where user\_id is the desired account. DELETes the account of the given user\_id.

- **Endpoint:** /user/delete

- **Method:** DELETE

- **Parameters:** user\_id (number)

- **Sample Request:**

```
http DELETE users.localhost/user/delete user_id=1
```

- **Success Response:**

```
HTTP/1.1 200 OK
content-length: 37
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:25:21 GMT
server: gunicorn/20.0.4

{
  "message": "User has been deleted."
}
```

- **Error Response:**

```
HTTP/1.1 404 Not Found
content-length: 86
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:25:46 GMT
server: gunicorn/20.0.4

{
  "message": "User you are trying to delete doesn't exist.",
  "status": "404 Not Found"
}
```

○ **new\_user(username, email, password, bio):**

- **Description:** Takes 4 parameters username, email, password, and bio, where username, email, password, and bio are the information for the new user. POSTs a new account with the given information.
- **Endpoint:** /user/new
- **Method:** POST
- **Parameters:** username, email, password, bio
- **Sample Request:**

```
http POST users.localhost/user/new/ username=Test
email=testemail@gmail.com password=test bio='wow so cool'
```

- **Success Response:**

```
HTTP/1.1 201 Created
content-length: 103
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:35:01 GMT
location: /user/5
server: gunicorn/20.0.4

{
  "bio": "wow so cool",
  "email": "testemail@gmail.com",
  "id": 5,
  "password": "test",
  "username": "Test"
}
```

- **Error Response:**

```
HTTP/1.1 409 Conflict
content-length: 78
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:53:43 GMT
server: gunicorn/20.0.4

{
  "message": "UNIQUE constraint failed: users.email",
  "status": "409 Conflict"
}
```

- **followers(username):**

- **Description:** Takes one parameter username, where username is the desired account. GETs the account information of the followers of the specified account.
- **Endpoint:** /user/<username>/followers
- **Method:** GET
- **Parameters:** username (text)
- **Sample Request:**

```
http GET users.localhost/user/ProfAvery/followers
```

- **Success Response:**

```
HTTP/1.1 200 OK
content-length: 36
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:55:33 GMT
server: gunicorn/20.0.4

[
  {
    "follower_user": "KevinAWortman"
  }
]
```

- **Error Response:**

```
HTTP/1.1 404 Not Found
content-length: 67
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:56:04 GMT
server: gunicorn/20.0.4

{
  "message": "ProfAver=y does not exist",
  "status": "404 Not Found"
}
```

○ **following(username):**

- **Description:** Takes one parameter username, where username is the desired account. GETs the account information of who the specified account is following.
- **Endpoint:** /user/<username>/following
- **Method:** GET
- **Parameters:** username (text)

- **Sample Request:** http GET users localhost/user/<username>/following

```
http GET users localhost/user/ProfAvery/following
```

- **Success Response:**

```
HTTP/1.1 200 OK
content-length: 70
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:57:31 GMT
server: gunicorn/20.0.4

[
  {
    "following_user": "KevinAWortman"
  },
  {
    "following_user": "Beth_CSUF"
  }
]
```

- **Error Response:**

```
HTTP/1.1 404 Not Found
content-length: 68
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 04:58:00 GMT
server: gunicorn/20.0.4

{
  "message": "ProfAveASry does not exist",
  "status": "404 Not Found"
}
```

- **follow(follower\_username, following\_username):**

- **Description:** Takes two parameters follower\_username, following\_username, where follower\_username is the account that wants to follow another, and following\_username is the account that is to be followed. POSTs the new connection between these accounts to the followers table.
- **Endpoint:** /user/follow
- **Method:** POST
- **Parameters:** follower\_username (text), following\_username (text)



■ **Sample Request:**

```
http POST users.localhost/user/follow follower_username=TestUser
following_username=ProfAvery
```

■ **Success Response:**

```
HTTP/1.1 201 Created
content-length: 46
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:01:27 GMT
server: gunicorn/20.0.4

{
  "follower_id": 4,
  "following_id": 1,
  "id": 6
}
```

■ **Example Error Response:**

```
HTTP/1.1 409 Conflict
content-length: 69
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:02:13 GMT
server: gunicorn/20.0.4

{
  "message": "You already follow ProfAvery",
  "status": "409 Conflict"
}
```

○ **unfollow(follower\_username, following\_username):**

- **Description:** Takes two parameters follower\_username, following\_username, where follower\_username is the account that wants to follow another, and following\_username is the account that is to be followed. DELETes the connection between these accounts from the followers table.
- **Endpoint:** /user/unFollow
- **Method:** DELETE
- **Parameters:** follower\_username, following\_username

■ **Sample Request:**

```
http DELETE users.localhost/user/unFollow follower_username=TestUser
following_username=ProfAvery
```

■ **Success Response:**

```
HTTP/1.1 201 Created
content-length: 49
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:04:53 GMT
server: gunicorn/20.0.4

{
  "message": "TestUser has un-followed ProfAvery"
}
```

■ **Error Response:**

```
HTTP/1.1 404 Not Found
content-length: 88
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:07:09 GMT
server: gunicorn/20.0.4

{
  "message": "User you are trying to unFollow does not exist",
  "status": "404 Not Found"
}
```

## ● Timeline Microservice:

### ○ public\_timeline():

- **Description:** Takes no parameters. GETs the posts of all users and lists them in descending chronological order by timestamp.
- **Endpoint:** /posts/all
- **Method:** GET
- **Parameters:** username (text)
- **Sample Request:**

```
http GET timelines.localhost/posts/all
```

### ■ Success Response:

```
HTTP/1.1 200 OK
content-length: 1728
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:14:44 GMT
server: unicorn/20.0.4

[
  {
    "id": 7,
    "original_post_url": "http://timelines.localhost/posts/6",
    "repost": 1,
    "text": "#cpsc315 #engr190w NeurIPS is $25 for students and $100 for
non-students this year!
https://medium.com/@NeurIPSCnf/neurips-registration-opens-soon-6711158
1de99",
    "timestamp": "2021-10-23 04:52:13",
    "username": "TestUser"
  },
  {
    "id": 6,
    "original_post_url": null,
    "repost": 0,
    "text": "#cpsc315 #engr190w NeurIPS is $25 for students and $100 for
non-students this year!
https://medium.com/@NeurIPSCnf/neurips-registration-opens-soon-6711158
1de99",
    "timestamp": "2021-10-23 04:52:13",
    "username": "Beth_CSUF"
  }
]
```

```

    "id": 5,
    "original_post_url": null,
    "repost": 0,
    "text": "I keep seeing video from before COVID, of people not
needing to mask or distance, and doing something like waiting in line at
Burger King. YOU'RE WASTING IT!",
    "timestamp": "2021-10-23 04:52:12",
    "username": "KevinAWortman"
  },
  {
    "id": 4,
    "original_post_url": null,
    "repost": 0,
    "text": "If academia were a video game, then a 2.5 hour administrative
meeting that votes to extend time 15 minutes is a fatality. FINISH HIM",
    "timestamp": "2021-10-23 04:52:12",
    "username": "KevinAWortman"
  },
  {
    "id": 3,
    "original_post_url": null,
    "repost": 0,
    "text": "Yes, the header file ends in .h. C++ is for masochists.",
    "timestamp": "2021-10-23 04:52:12",
    "username": "ProfAvery"
  },
  {
    "id": 2,
    "original_post_url": null,
    "repost": 0,
    "text": "FYI: https://www.levels.fyi/still-hiring/",
    "timestamp": "2021-10-23 04:52:12",
    "username": "ProfAvery"
  },
  {
    "id": 1,
    "original_post_url": null,
    "repost": 0,
    "text": "Meanwhile, at the R1 institution down the street...
https://uci.edu/coronavirus/messages/200710-sanitizer-recall.php",
    "timestamp": "2021-10-23 04:52:12",
    "username": "ProfAvery"
  }
]

```

■ **Error Response:**

NO ERROR RESPONSE

○ **user\_public\_timeline(username):**

- **Description:** Takes one parameter username, where username is the desired account. GETs the posts a user with “username” has posted and lists them in descending chronological order by timestamp.
- **Endpoint:** /<username>/posts
- **Method:** GET
- **Parameters:** username (text)
- **Sample Request:**

```
http GET timelines.localhost/ProfAvery/posts
```

■ **Success Response:**

```
HTTP/1.1 200 OK
content-length: 584
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:09:31 GMT
server: gunicorn/20.0.4

[
  {
    "id": 1,
    "original_post_url": null,
    "repost": 0,
    "text": "Meanwhile, at the R1 institution down the street...
https://uci.edu/coronavirus/messages/200710-sanitizer-recall.php",
    "timestamp": "2021-10-23 04:52:12",
    "username": "ProfAvery"
  },
  {
    "id": 2,
    "original_post_url": null,
    "repost": 0,
    "text": "FYI: https://www.levels.fyi/still-hiring/",
    "timestamp": "2021-10-23 04:52:12",
    "username": "ProfAvery"
  },
  {
    "id": 3,
    "original_post_url": null,
    "repost": 0,
    "text": "Yes, the header file ends in .h. C++ is for masochists.",
    "timestamp": "2021-10-23 04:52:12",
    "username": "ProfAvery"
  }
]
```

■ **Error Response:**

```
HTTP/1.1 404 Not Found
content-length: 66
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:10:52 GMT
server: gunicorn/20.0.4

{
  "message": "ProfAvey does not exist.",
  "status": "404 Not Found"
}
```

○ **user\_home\_timeline(auth\_user, username)**

- **Description:** Takes two parameters auth\_user and username, where auth\_user is the authorized user and username is the entered username. The authorized user is checked with “username”. GETs the posts by all users that this user follows and lists them in descending chronological order by timestamp. Uses Hug Basic Authentication to validate the user.
- **Uses basic authentication:** username, password
- **Endpoint:** /<username>/home
- **Method:** GET
- **Parameters:** username (text), basic authentication
- **Sample Request:**

```
http -a ProfAvery:password timelines.localhost/ProfAvery/home
```

■ **Success Response:**

```
HTTP/1.1 200 OK
content-length: 829
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:18:12 GMT
server: gunicorn/20.0.4
www-authenticate:

[
  {
    "id": 6,
    "original_post_url": null,
    "repost": 0,
    "text": "#cpsc315 #engr190w NeurIPS is $25 for students and $100 for non-students this year! https://medium.com/@NeurIPSConf/neurips-registration-opens-soon-6711158"
```

```

1de99",
  "timestamp": "2021-10-23 04:52:13",
  "username": "Beth_CSUF"
},
{
  "id": 4,
  "original_post_url": null,
  "repost": 0,
  "text": "If academia were a video game, then a 2.5 hour administrative
meeting that votes to extend time 15 minutes is a fatality. FINISH HIM",
  "timestamp": "2021-10-23 04:52:12",
  "username": "KevinAWortman"
},
{
  "id": 5,
  "original_post_url": null,
  "repost": 0,
  "text": "I keep seeing video from before COVID, of people not
needing to mask or distance, and doing something like waiting in line at
Burger King. YOU'RE WASTING IT!",
  "timestamp": "2021-10-23 04:52:12",
  "username": "KevinAWortman"
}
]

```

■ **Error Response:**

```

HTTP/1.1 401 Unauthorized
content-length: 101
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:19:13 GMT
server: gunicorn/20.0.4
www-authenticate: basic realm="simple"

{
  "errors": {
    "Invalid Authentication": "Provided Basic HTTP Authentication
credentials were invalid"
  }
}

```

○ **get\_post\_by\_id(id):**

- **Description:** Takes one parameter id, where id is the user id of the desired account. GETs the posts of a user with the desired user ID and lists them in descending chronological order by timestamp.
- **Endpoint:** /posts/id
- **Method:** GET

- **Parameters:** id (number)

- **Sample Request:**

```
http GET timelines.localhost/posts/1
```

- **Success Response:**

```
HTTP/1.1 200 OK
content-length: 238
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:20:11 GMT
server: gunicorn/20.0.4

{
  "id": 1,
  "original_post_url": null,
  "repost": 0,
  "text": "Meanwhile, at the R1 institution down the street...
https://uci.edu/coronavirus/messages/200710-sanitizer-recall.php",
  "timestamp": "2021-10-23 04:52:12",
  "username": "ProfAvery"
}
```

- **Error Response:**

```
HTTP/1.1 404 Not Found
content-length: 68
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:20:25 GMT
server: gunicorn/20.0.4

{
  "message": "post_id:234 does not exist",
  "status": "404 Not Found"
}
```

- **new\_post(auth\_user, text):**

- **Description:** Takes two parameters auth\_user and text, where auth\_user is the authorized user and text is the entered username. POSTs a new user with the desired username.
- **Uses basic authentication:** username, password
- **Endpoint:** /compose/post
- **Method:** POST
- **Parameters:** text(text), basic authentication
- **Sample Request:**



```
http -a ProfAvery:password POST http://timelines.localhost/compose/post
text="this is a test post boi"
```

■ **Success Response:**

```
HTTP/1.1 201 Created
content-length: 69
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:26:03 GMT
location: /posts/8
server: gunicorn/20.0.4
www-authenticate:

{
  "id": 8,
  "text": "this is a test post boi",
  "username": "ProfAvery"
}
```

■ **Error Response:**

```
HTTP/1.1 400 Bad Request
content-length: 62
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:27:02 GMT
server: gunicorn/20.0.4
www-authenticate:

{
  "errors": {
    "text": "Required parameter 'text' not supplied"
  }
}
```

○ **new\_repost(id, auth\_user):**

- **Description:** Takes two parameters id and auth\_user, where id is the entered user ID and auth\_user is the authorized user. POSTs a new user with the desired username.
- **Endpoint:** /repost/<id>
- **Method:** POST
- **Parameters:** auth\_user (text), text (text)
- **Sample Request:**

```
http -a ProfAvery:password POST timelines.localhost/repost/1
```

■ **Success Response:**

```
HTTP/1.1 201 Created
content-length: 237
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:29:16 GMT
location: /posts/9
server: gunicorn/20.0.4
www-authenticate:

{
  "id": 9,
  "original_post_url": "http://timelines.localhost/posts/1",
  "repost": true,
  "text": "Meanwhile, at the R1 institution down the street...
https://uci.edu/coronavirus/messages/200710-sanitizer-recall.php",
  "username": "ProfAvery"
}
```

■ **Error Response:**

```
HTTP/1.1 409 Conflict
content-length: 76
content-type: application/json; charset=utf-8
date: Sat, 23 Oct 2021 05:29:33 GMT
server: gunicorn/20.0.4
www-authenticate:

{
  "message": "You have already reposted this post",
  "status": "409 Conflict"
}
```